NORTHERN ARIZONA UNIVERSITY College of Engineering, Forestry, and Natural Sciences

Group:	James Beasley, Charles Beck, Charles Duso, Alexander Grzesiak, Erik Strauss
Project Title:	Boston University - Microfluid Experimentation Data Generator
Deliverable:	D.1.2. Consumer Discovery
Course:	CS386 – Spring 2017
Instructor:	Professor Gerosa
Github:	https://github.com/TheAwesomeEgg/CS386ProjectGroup1.git

Value Proposition

Our web application is to be used in research towards microfluidic experimentation being conducted at Boston University. The application will allow researchers at Boston University to easily communicate instructions to the hardware they are developing for experimentation. At the core of the application it will be a web UI that allows users to create data structures, and containing processes that will be executed by hardware. By introducing a selection interface that allows the user to choose the hardware they wish to input to and the instructions they would like the hardware to perform, this product allows for more in-depth approaches to microfluidic experiments.

Consumer Segment

Since our web application is specific to Boston University there are few key groups that could find use out of it. These groups would need to be other academic research facilities that are carrying out the same experiments as Boston University. There for, with in BU there is the primary group of consumers(users) of the application that include the research team and the professors involved with the microfluidic research. The second group we see having an interest for this application are other prospective students within the school who want to engage in the experiments but who are not yet a part of the research team.

Interviews

Taylor Walker Interview Questions

1. When searching for new software to assist you in your work are there some primary features you look for, such as pleasing aesthetics, apparent ease of use, or good online documentation?

   Visually pleasing is pretty important for me, or options to change the aesthetics, such as "light" or "dark" modes on sites or text editors such as Atom.

The other main feature for me is intuitive design and ease of use. I like software where I don't have to think, I can click around a bit and be able to figure out how to do something quite easily and intuitively.

Online documentation is not as important for me, because I can normally google and find answers/solutions somewhere.

2. Do you have a preference between desktop applications versus web applications? If you do prefer one over the other, why?

   I prefer desktop applications, they are normally more responsive and don't clock my internet or tabs as much. I just feel like desktop applications are normally better designed, or have more features.

3. For desktop applications what platforms do you like to see supported?

   I like Android phones, and have a Mac for a laptop. So, I like to see those platforms supported.

4. What platform do you perform most of your work on? Have you ever had issues with software not supporting your platform or not being optimized for your screen size / resolution?

   I perform most of my work on my Mac laptop, and it seems like most software has both Mac and Windows options, so I don't have too much of a problem with that. Sometimes software is made specifically for Mac, or specifically for Windows, which is also fine, because there always seems to be a Mac only software that would be comparable to any Windows only software.

   I would rather have a software be optimized for my OS, or not have it made at all, thus forcing me to find something better that is.

   Many games are not optimized for Macs, sometimes they try to make a game crossover to Mac but do it poorly, which is sometimes more frustrating then just being told that they don't have a Mac version.

5. For web applications, do you prefer client side or server side scripting? Or is this issue generally not important to you?

   I think it more applies to the company programing it. It's not as important to me as a user.

6. How important is good code documentation to you? If the software you are using does not contain a feature you desire, are you likely to look into the source code and consider adding it?

   I'm not normally the type of person to look into software's code documentation and mess around with it. So, while I appreciate good code documentation as a computer science student, it is not necessarily important to me when considering a software.

7. How important are aesthetics in software to you? Would you ever choose a competing piece of software because it is more visually pleasing despite having less features?

   Yes. Maybe I am just a visual person, but visually pleasing sets a "tone" while using the software. Even when visiting websites, there could be great content on one versus another less informative one. But if the less informative one looks and feels more professional, I am more apt to use it.

Shane McCormack Interview Questions

1. What is the Neptune project?

   A simple user-interface to aid microfluidic creation from design to control of the fabricated device

2. What is your role within the Neptune project?

   I design simplified user input, as well as convert this level of abstraction to code used by CIDAR lab's Place and Route algorithm for drawing the microfluidic design.

3. Currently, how do you write instructions for the hardware being developed?

   The current control over the microfluidic chips is limited to manual real-time valve control via clicking on the desired valve on the UI. There is no current method of writing instructions for the new hardware models, however their respective developers are writing control APIs that Neptune will leverage.

4. What is the primary desire for the software being requested? Is it intended to save time? Is it to allow more people the ability to write instruction sets for the hardware?

   To provide an simple and easy to use user interface page for the creation of automated instruction for the new hardware modules

5. How important is it to you that most of the scripting be handled server side as opposed to client side or vice versa? Is it not an important issue?

   I don't believe this is an important issue.

6. How important is readability inside of the generated instruction files? Should they be readable and modifiable outside of the software?

   I would like to be able to upload previously created instruction files, but at the moment my plan is to develop a simple json format.

7. Are there any popular file formats that the software should have the ability to write to? Should the software have support for multiple file types?

   JSON. No.

8. How important are the aesthetics of the software? Should the software carry the branding of the project?

   Pretty important. This will be what the user interacts with, so we'd like it to look nice. As a tool, Neptune has a logo/color scheme for most pages that can be kept for this page.

9. How important is modularity in the software? Is it possible that new types of hardware support will need to be added to the software in the future?

   Very important. There will almost certainly be new hardware modules/new instructions for old hardware modules that need to be added. The ease of adding new instructions/hardware modules into the control user input page is very important.

10. Do you see other possible applications of this software to future projects? If so, what features would you want built in to give the software more value?

    Not at the moment. Possibly as a skeleton for designing a drag-and-drop version of my current structural user input in the future.

Brad Gibbons Interview Questions

1. When searching for new software to assist you in your work are there some primary features you look for, such as pleasing aesthetics, apparent ease of use, or good online documentation?

   When working with new software, online documentation is my number one feature. I love being able to look up libraries and functions and seeing a comprehensive resource of what the language offers. This is why I love using Java so much.

2. Do you have a preference between desktop applications versus web applications? If you do prefer one over the other, why?

   I prefer desktop application because I tend to have a lot of windows open at a time, and having a unique process allows me to be able to easily manage everything I am working on.

3. For desktop applications what platforms do you like to see supported?

   I like to see mostly Microsoft Windows and Linux platforms supported. I have a desktop at home which runs Windows, and a laptop that runs Arch Linux. My ideal applications have support with both platform.

4. What platform do you perform most of your work on? Have you ever had issues with software not supporting your platform or not being optimized for your screen size / resolution?

   I do most of my work on my Windows machine since I have 3 monitors and can easily have multiple windows open. I haven't had many issues with my platform not being

supported since most of the programs I've used have been built specifically for Windows running at 1920x1080 resolution.

5. For web applications, do you prefer client side or server side scripting? Or is this issue generally not important to you?

   Currently this is not an issue that is important to me since I haven't worked much with web applications.
6. How important is good code documentation to you? If the software you are using does not contain a feature you desire, are you likely to look into the source code and consider adding it?

   Code documentation is a must for me. Even when writing my own programs, I keep comprehensive documentation. Currently, I do not go through source code of software and look to add my own features, but would consider this as I become more familiar with software development.

7. How important are aesthetics in software to you? Would you ever choose a competing piece of software because it is more visually pleasing despite having less features?

   I believe aesthetics is essential in software. Having a visually appealing application can often make the program easier to use. If the program has a ton of features, but is not easy to use, many of those features will go to waste and therefore I'll most likely use the visually appealing software.

Anish Asthana Interview Questions

1. When searching for new software to assist you in your work are there some primary features you look for, such as pleasing aesthetics, apparent ease of use, or good online documentation?

   Good Documentation, should accomplish what I am trying to do. I don't want to have to wrestle with the software to do what I want. *cough* Windows 10 *cough*

2. Do you have a preference between desktop applications versus web applications? If you do prefer one over the other, why?

   Depends on the situation. Web apps are better since I don't have to download anything in my computer, but if I have a spotty connection I'd prefer the desktop app. At the end of the day though, I would prefer the best functioning app. So if it works better on desktop, I will probably use the desktop app.

3. For desktop applications what platforms do you like to see supported?

   Windows, Linux

4. What platform do you perform most of your work on? Have you ever had issues with software not supporting your platform or not being optimized for your screen size / resolution?

   Windows, Linux. And Yup! Tons of shit doesn't work on Linux, so I still use Windows (games included).

5. For web applications, do you prefer client side or server side scripting? Or is this issue generally not important to you?

   No real preference either way. Just work well.

6. How important is good code documentation to you? If the software you are using does not contain a feature you desire, are you likely to look into the source code and consider adding it?

   Good documentation is important so that if something isn't working as expected I can understand why. I will consider that based on quality of source code, and effort on my part to add it.

7. How important are aesthetics in software to you? Would you ever choose a competing piece of software because it is more visually pleasing despite having less features?

   Nope